6-1-2012

The Intersection between Science and Computer Science is Almost Empty

Dick Hamlet
Portland State University

The Intersection of Computer 'Science' and Science is almost Empty

Dick Hamlet

Department of Computer Science
Portland State University
Portland, OR 97207
hamlet@cs.pdx.edu

Portland State
UNIVERSITY

Science, Engineering, Mathematics

- Categories for carving up knowledge (the world? reality?)
- Distinct goals, methods, philosophies
- Categorizing a particular discipline (like ‘molecular biology’) helps to understand it
- Misclassifying a discipline leads to confusion

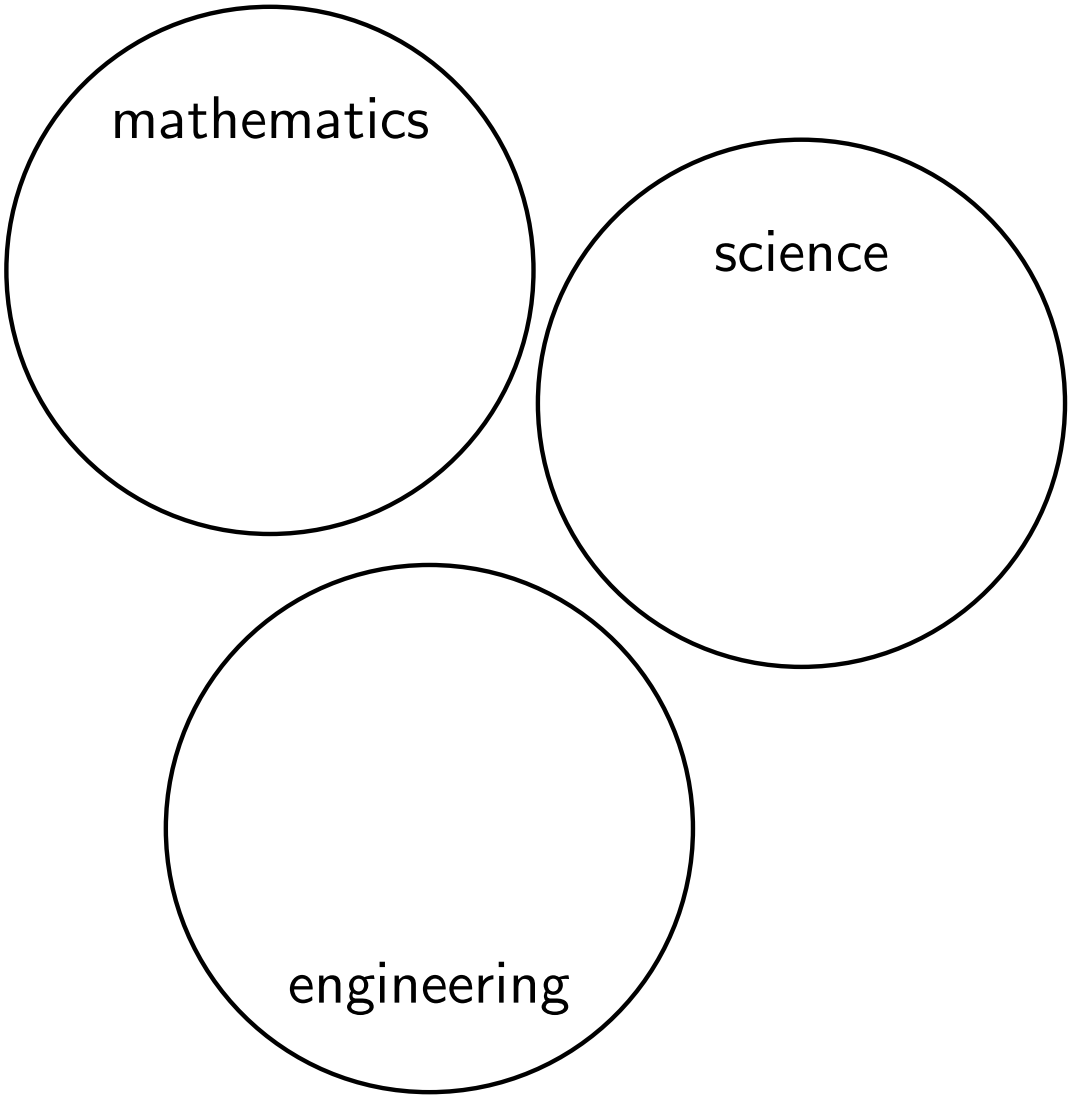
Science, Engineering, Mathematics

- Categories for carving up knowledge (the world? reality?)
- Distinct goals, methods, philosophies
- Categorizing a particular discipline (like 'molecular biology') helps to understand it
- Misclassifying a discipline leads to confusion

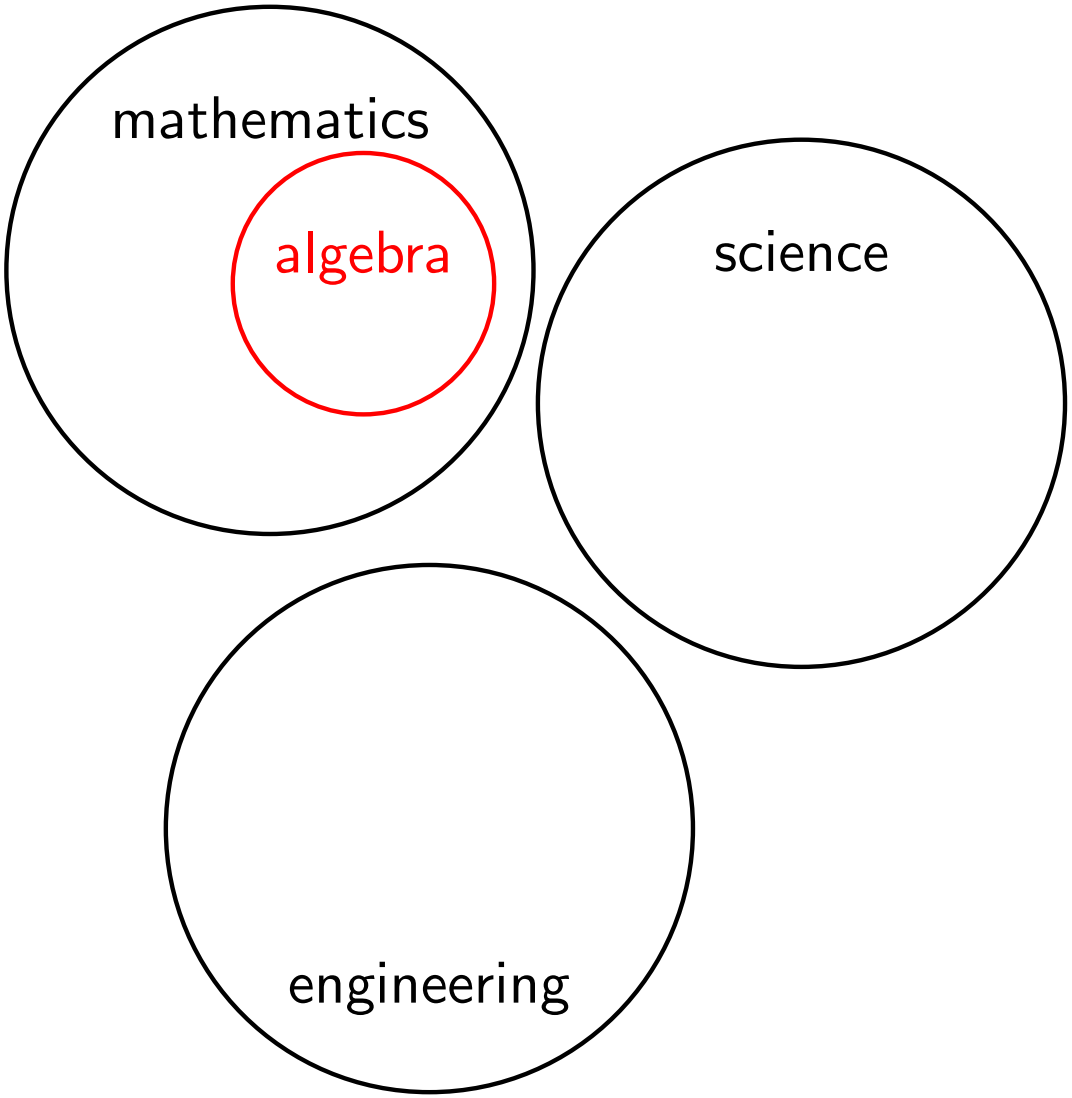
(Other important categories, like 'religion', 'economics', etc. omitted, along with more confusion)

Definitions to follow...

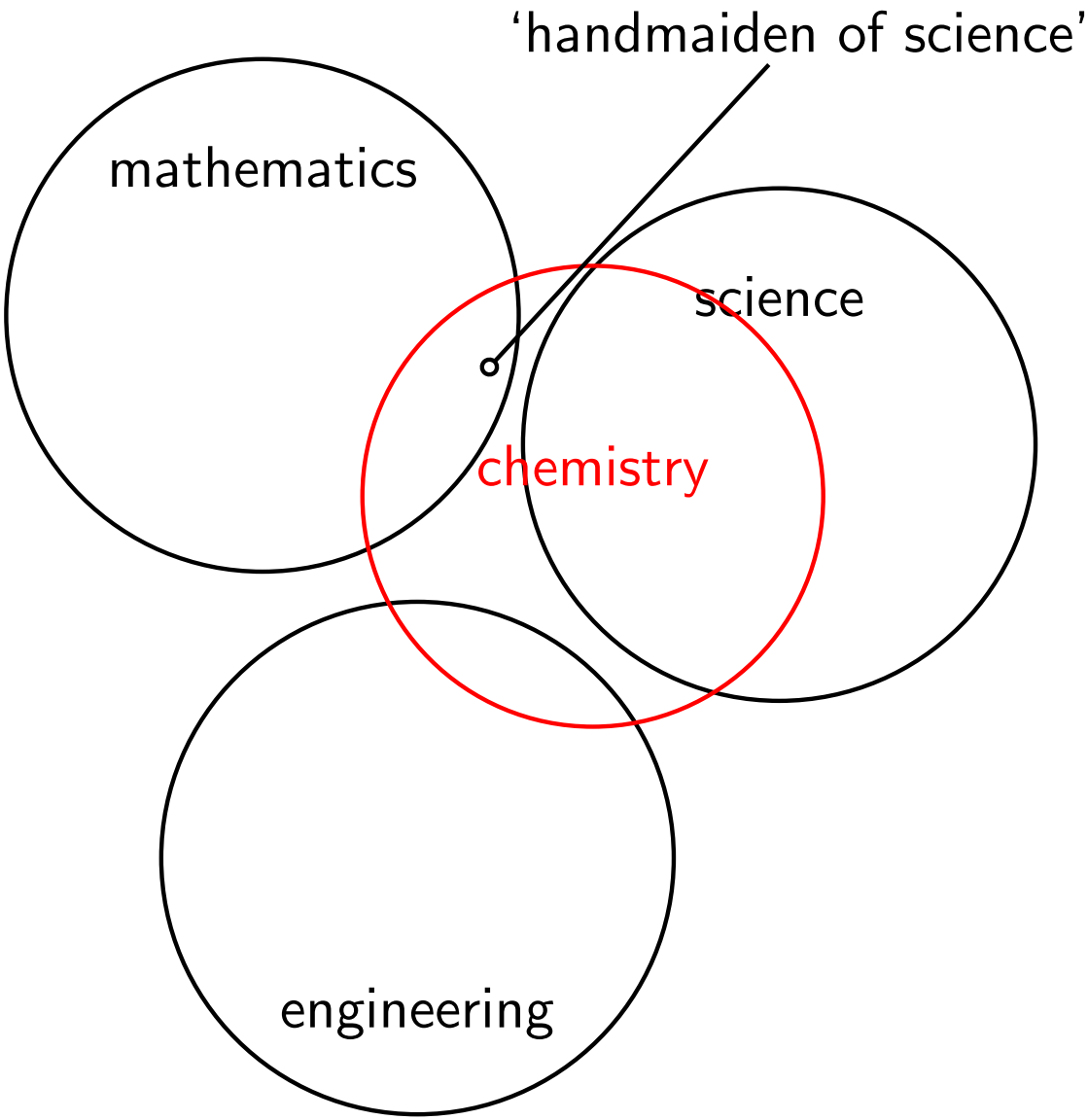
Classification Examples



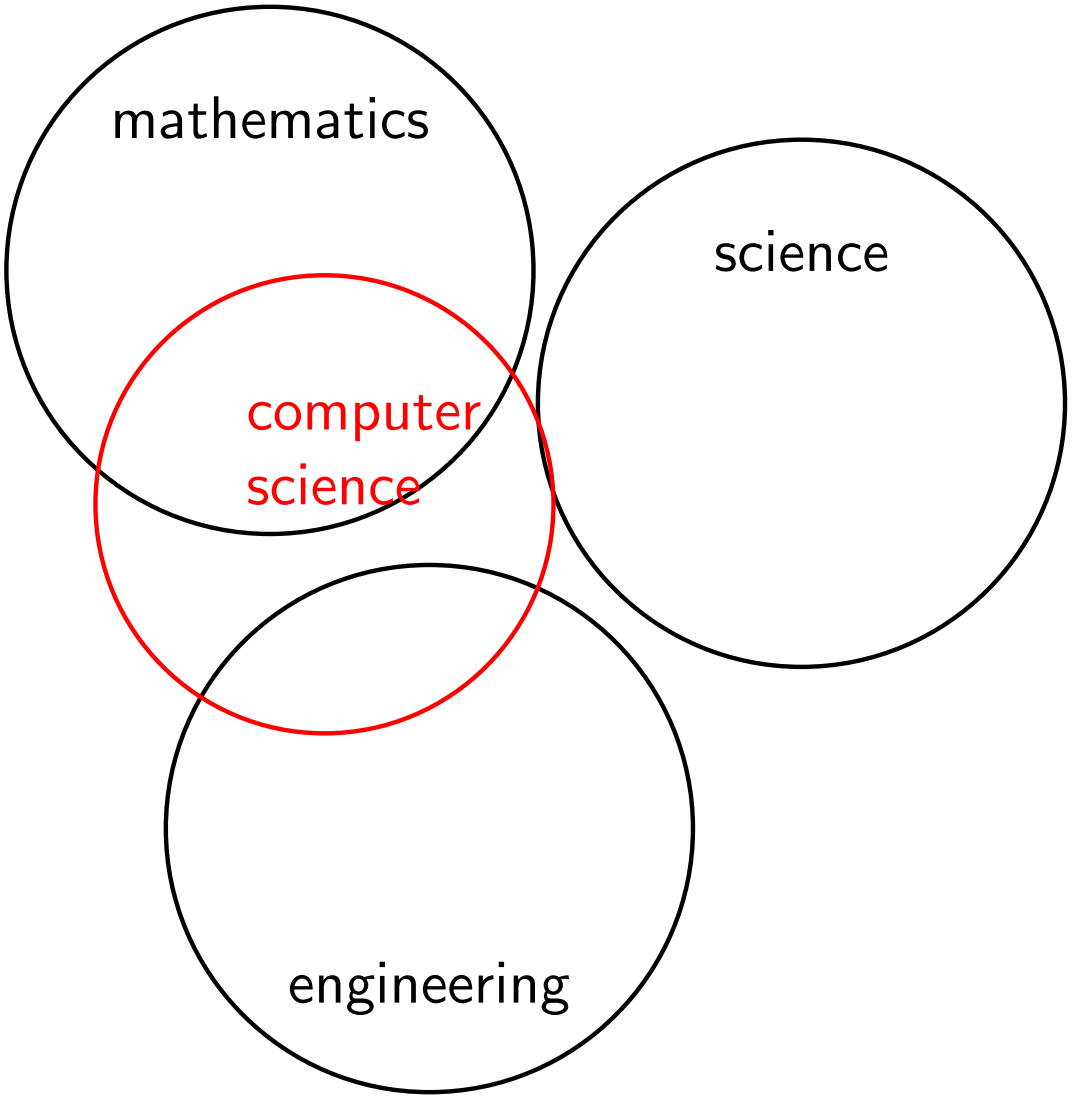
Classification Examples



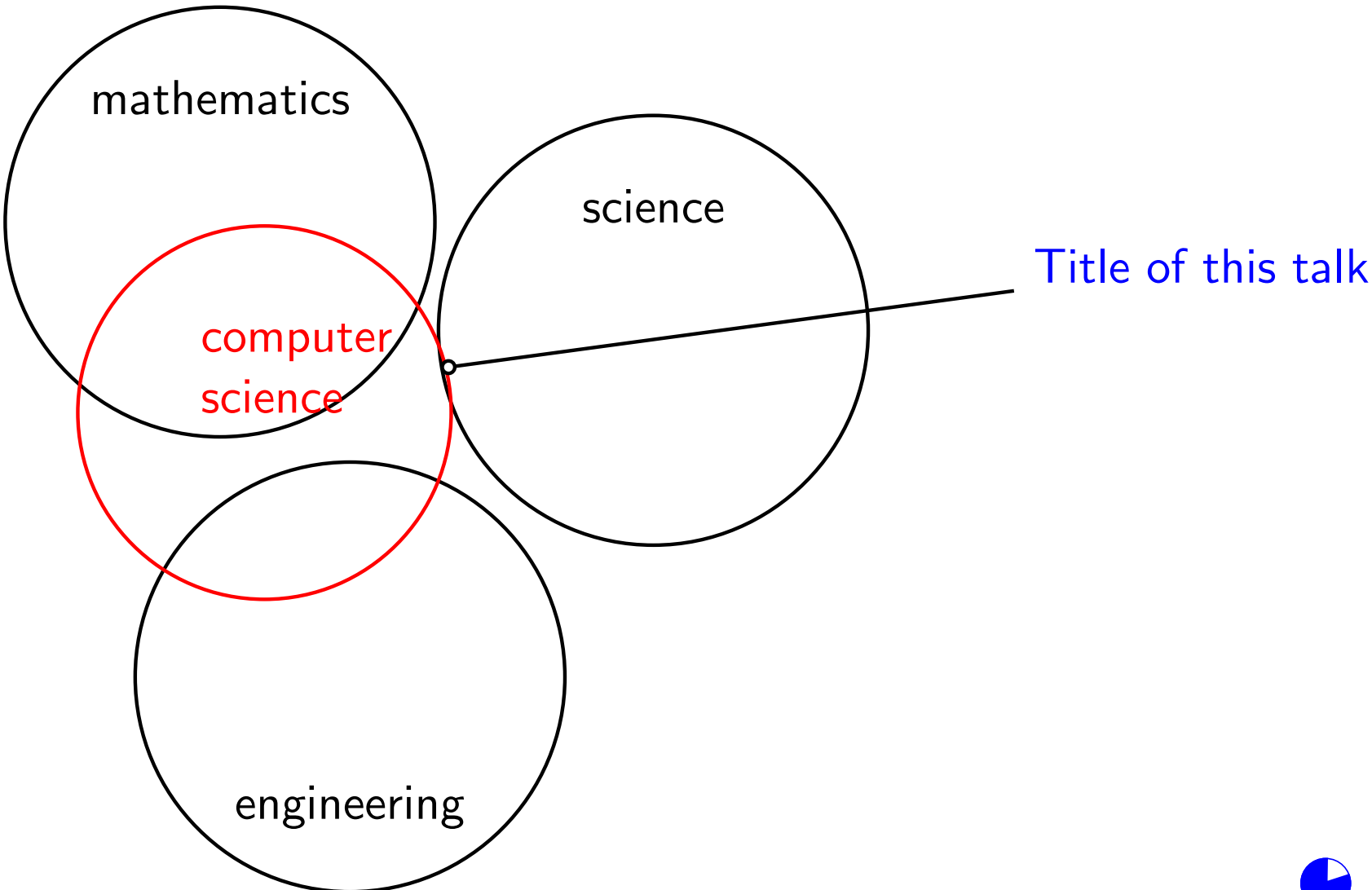
Classification Examples



Classification Examples



Classification Examples



Mathematical Theories

A body of:

- Definitions
- Axioms
- Theorems

Theorems are *proved* from Axioms using the shorthand of definitions.

Mathematical Theories

A body of:

- Definitions
- Axioms
- Theorems

Theorems are *proved* from Axioms using the shorthand of definitions.

> *"Mathematics may be defined as the subject in which we never know what we are talking about, nor whether what we are saying is true" - Bertrand Russell*

Engineering

Pessimist: “The glass is half empty.”

Optimist: “The glass is half full.”

Engineering

Pessimist: "The glass is half empty."

Optimist: "The glass is half full."

Engineer: "You need a smaller glass."

Engineering

Pessimist: "The glass is half empty."

Optimist: "The glass is half full."

Engineer: "You need a smaller glass."

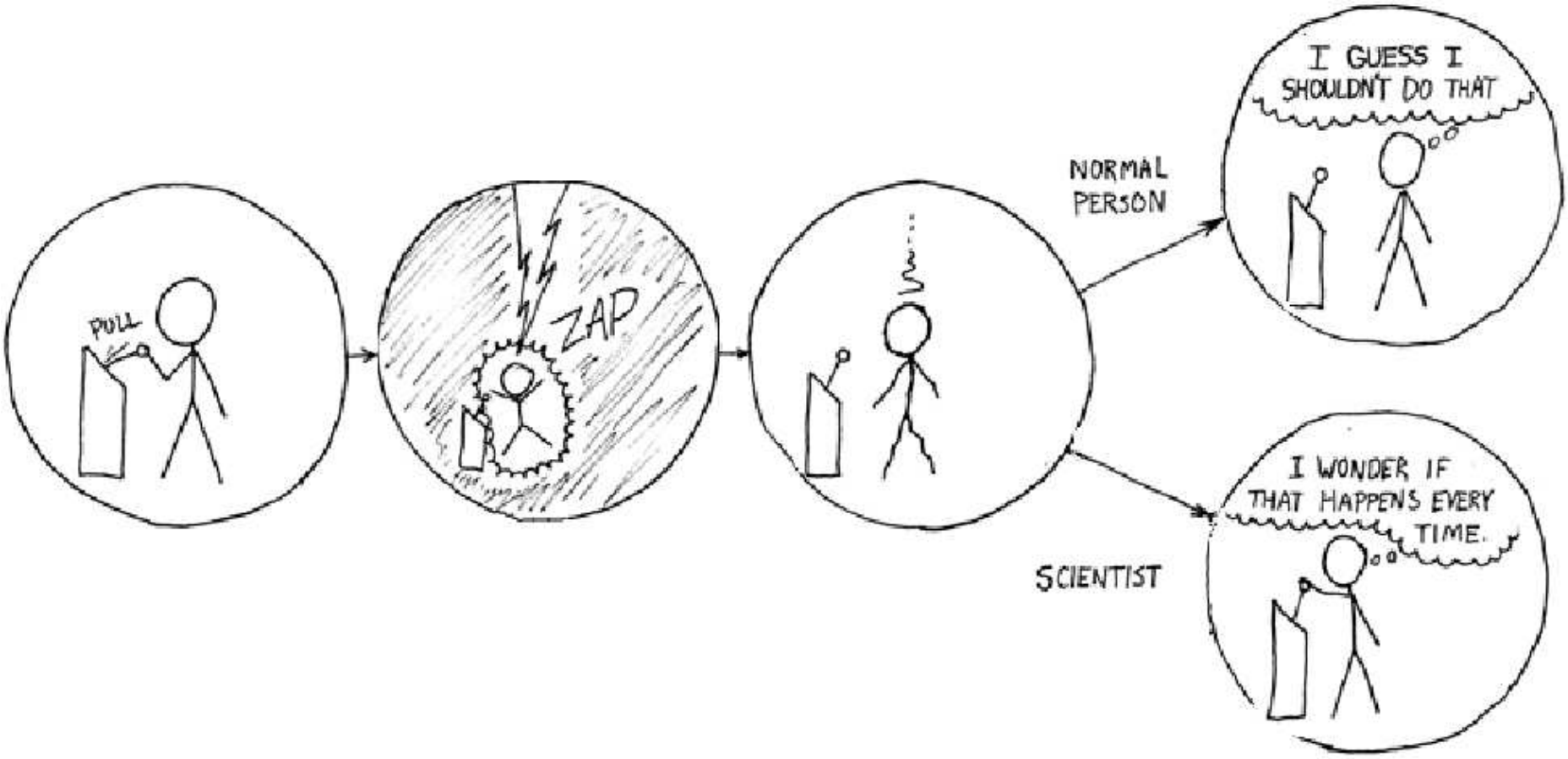
to engineer: *To arrange, manage or carry through by skillful or artful contrivance.*

Engineering Theory

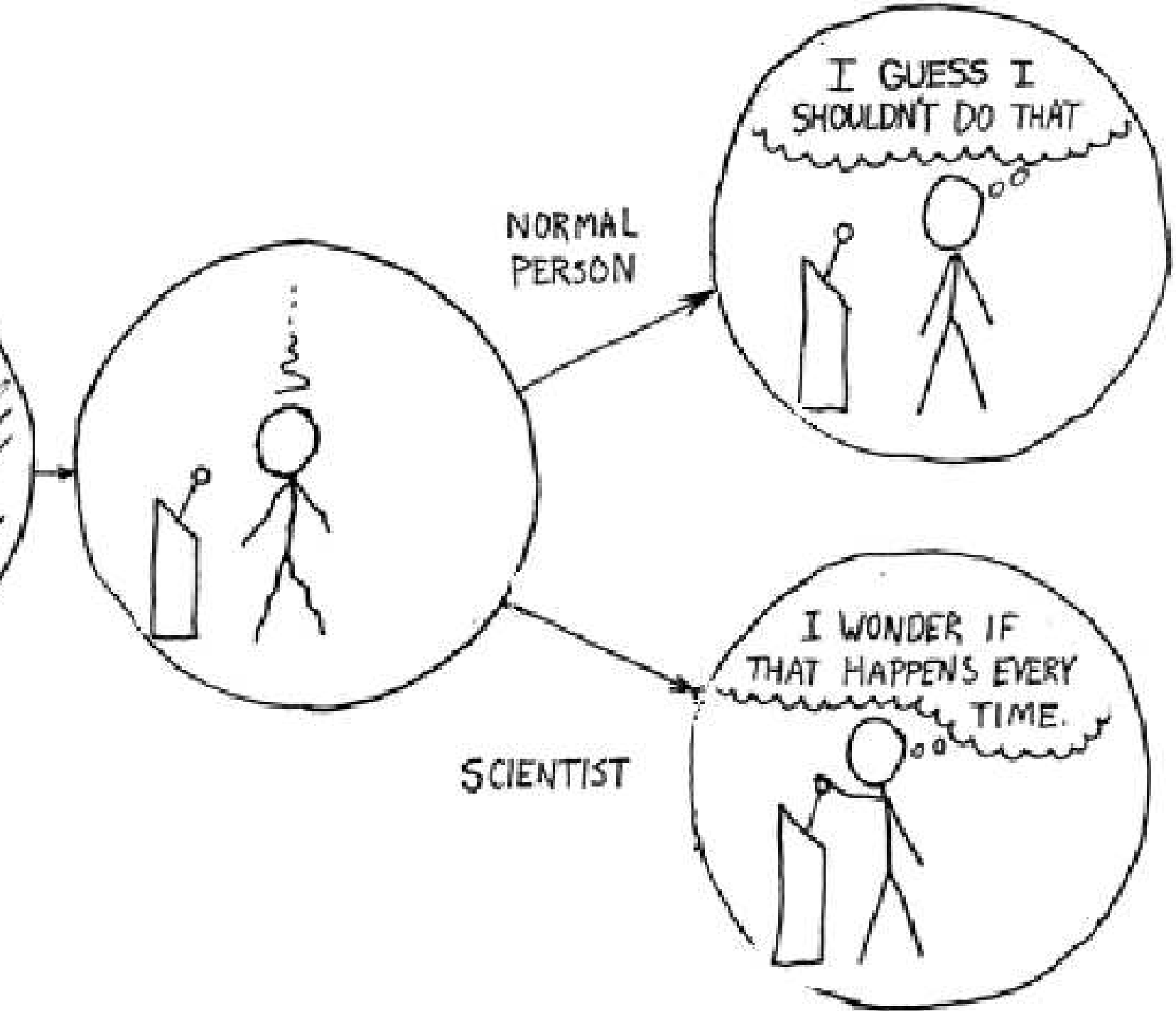
Engineers work day-to-day using *design rules*, abstract, detailed descriptions of how to 'carry through by skillful contrivance'

- Design rules must be routinely *usable* – minimal creativity needed to follow them
- Design rules must *work* – but by including a 'safety factor' they don't have to work perfectly
  - Example: Design of 5th-century BCE Greek temples: Use 'harmonious' ratios of dimensions for column spacing (but add a few extra columns)
  - Example: Design of steel-frame skyscrapers before 1950s: design the frame assuming elastic deformation (but use some 'redundant' beams)

Science according to XKCD



(adapted from `http://xkcd.com/242`)


I GUESS I SHOULDN'T DO THAT
NORMAL PERSON
I WONDER IF THAT HAPPENS EVERY TIME.
SCIENTIST

Science

The ‘featherless biped’ style of definition

A discipline for describing the world that strives for:

- generality
- originality
- unification
- formality
- surprise
- etc.

Science

The ‘featherless biped’ style of definition

But other disciplines are not excluded, notably religion and economics

Diogenes’s response to Plato’s definition of man as a featherless biped was this “man”:

Karl Popper's Science

- Operationally defined physics (Bridgman, 1928): A concept must be *measurable*
- A scientific theory must be *falsifiable*
  There must be observations to test the theory, and one possible outcome must be that the theory is *wrong*
  In a dispute, each side stipulates an experiment that will prove *the other side* right

  > *"I may be wrong and you may be right, and by an effort, we may get nearer to the truth." - (1959)*

Karl Popper's Science

- Operationally defined physics (Bridgman, 1928): A concept must be *measurable*
- A scientific theory must be *falsifiable*
  There must be observations to test the theory, and one possible outcome must be that the theory is *wrong*
  In a dispute, each side stipulates an experiment that will prove *the other side* right

  > *"I may be wrong and you may be right, and by an effort, we may get nearer to the truth." - (1959)*

- The best a scientific theory can aspire to is "not (yet) disproved"
- *Mathematics is not science* by Popper's definition

The Three Kinds of 'Theory'

- **Mathematical:** Definitions, axioms, theorems, proof
- **Engineering:** Design rules with safety factors
- **Scientific:** Falsifiable statements about observable objects

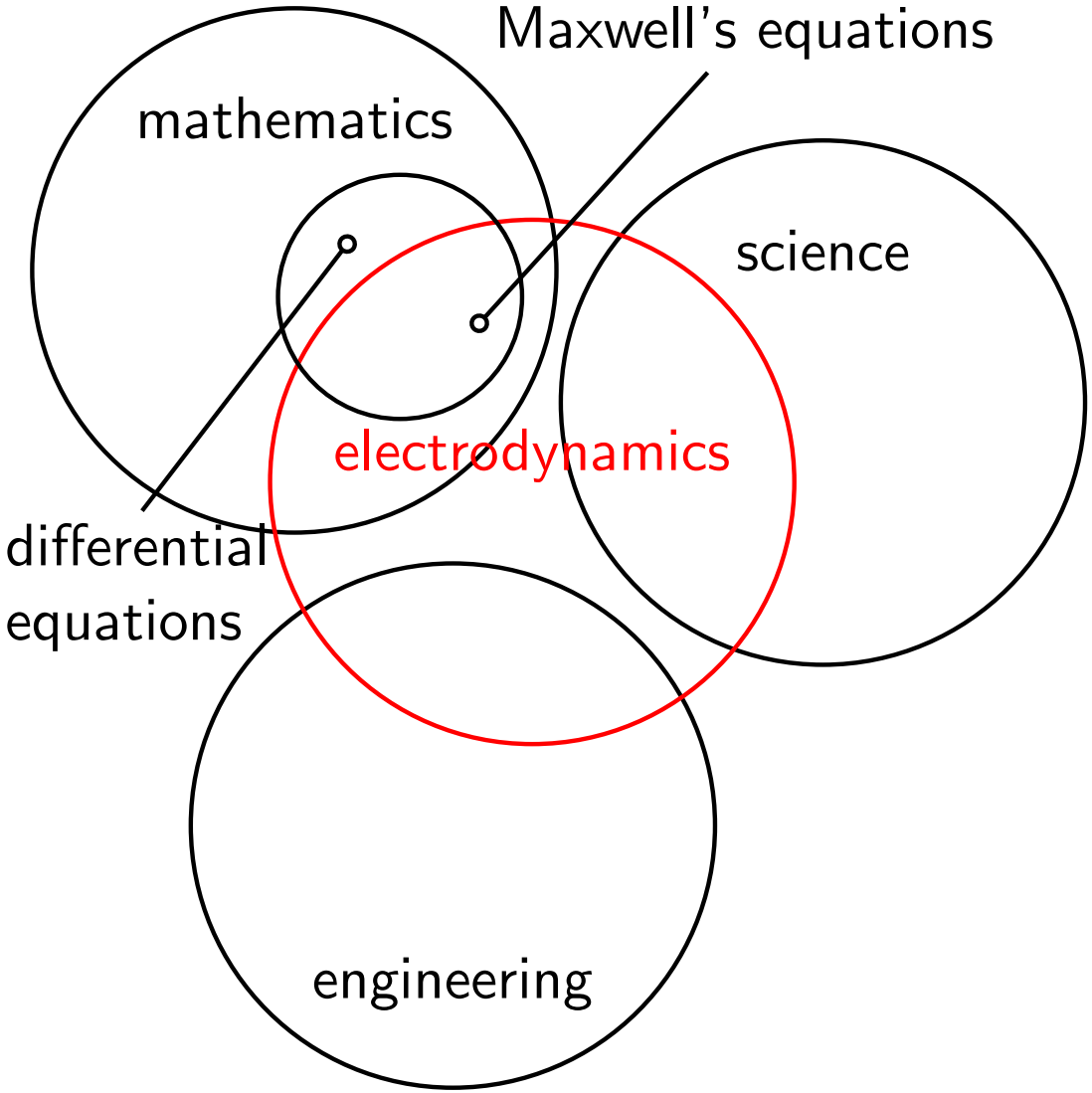
Engineering design rules are based on scientific theory with safety factors to cover mistakes in the science

The role of mathematics is more complicated...

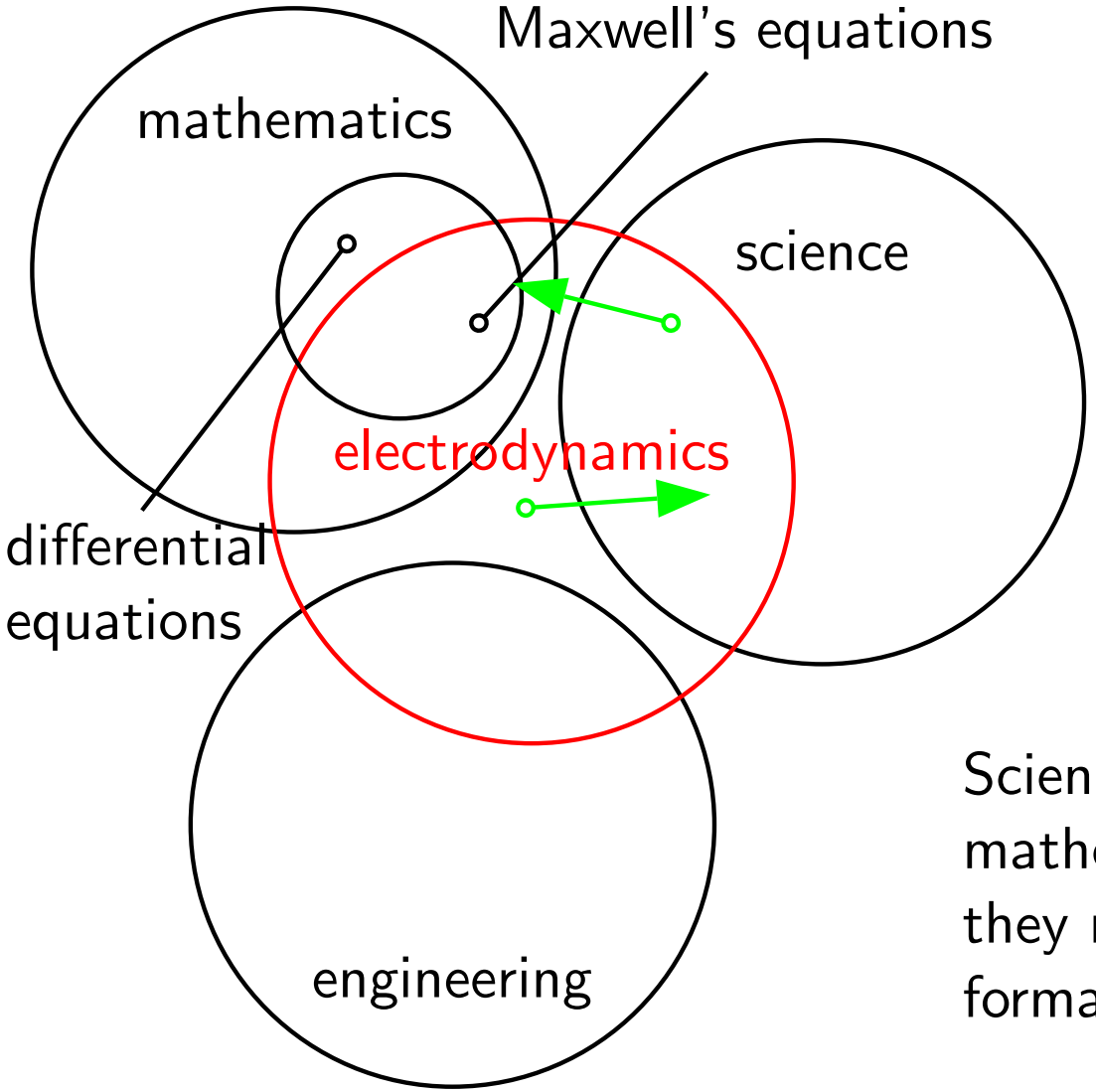
Applied Mathematics

- Correspondence between a mathematical theory (abstract) and real-world objects (observable)
- Axioms correspond to real-world statements that can be verified by observation (experiments)
- Theorems correspond to real-world statements that must therefore be observably correct
- The virtue in this exercise is that a deep theorem may yield a significant real-world insight
- The drawback is that 'verification' can never be complete

Example: Electronic Disciplines



Example: Electronic Disciplines



Science and applied mathematics are greedy; they measure and formalize.

Computer Science?

The central questions:

- What is a program?
- What are programs' 'applied mathematics'?

There are two possibilities:

1. A program is a real-world object for experimental observation

   If the axioms of an applied mathematical theory of programs fail in the real world, the only remedy is to alter the theory or the mapping. The world is not available for revision.

2. A program is a mathematical object – an abstract definition

   If the axioms of another mathematical theory fail in the program theory, *either* theory can be changed to fit. One theory merely explains the other.

So which is it?

Consider 'Debugging' a Program

- A programmer seeks correspondence between a mathematical specification of a problem solution and a program.
- If the program doesn't 'work', either it or the specification is adjusted until they match (which is established by a mathematical proof).

Where is the 'real world' is this?

Programs are Mathematically Defined

Some program properties and their definitions:

- *syntax*: Formal grammar (Chomsky, Backus, Naur, et al.)
- *semantics*: Fixpoint denotational (Dijkstra, Mills, et al.)
- *runtime*: Computational complexity (Ritchie, Hartmanis, et al.)
- *language*: Gödel numbering (Turing, Church, et al.)

...and many others.

A program has no existence except as defined – messy, intractable mathematics is still mathematics

The Authorities Speak: C.A.R. Hoare

(The inventor of a logic for proving programs correct)

"Almost anything [dreadful] in software can be implemented, and even used, given enough determination."

"...[P]rograms themselves, as well as their specifications, are mathematical expressions."

"...this final assurance [that we know a programming 'law' is right] can in principle be given by mathematical reasoning and proof."

The Authorities Speak: Juris Hartmanis

(One of the founders of computational complexity)

> *"[Computer science] differs so basically from the other sciences that it has to be viewed as a new species among the sciences..."*
>
> *"...theories do not compete ... for which better explains the fundamental nature of information. Nor are new theories developed to reconcile theory with experimental results..."*
>
> *"...there are no experiments ... which could resolve ... problems [like $P = NP?$]..."*
>
> *"[Computer science experiments are] dramatic demos to show the possibility to do what was thought to be impossible."*

No Science $\Rightarrow$ Fewer Experiments

The Thesis Defense of Susan Theerie A comedy in one short act.

(SETTING: An academic conference room. SUSAN and her advisor are seated at a table with two other computer science professors, including MARV.)

MARV (steepling fingers): You seem to have some interesting theoretical ideas, although I didn't have time to read your math carefully. Have you implemented them to check your theorems?

SUSAN: No I haven't. It seemed silly to waste a whole year writing a program. It's better to check the proofs.

MARV: Too bad; this isn't a math department, you know.

SUSAN (collecting her papers): Hmmmm...

(BLACKOUT.)

No Science $\Rightarrow$ Fewer Experiments

Dear Dr. Turing:

Thank you for submitting your paper entitled "On computable numbers, with an application to the entscheidungsproblem." The referees are agreed that it has some merit, but would be much improved by experimentally testing your ideas.

Please resubmit your paper after you have constructed one of these 'machines' and can report on experiments with it. It would also help if you could come up with a catchy name for them. Might we suggest: 'shifting and printing tape device'; or, (although it redefines the word): 'computer'?

Sincerely yours,

Proceedings Editor

London Mathematical Society

No Science $\Rightarrow$ Better Mathematics

Programming language issues:

- Why does every language allow buffer overflow?
- Why is there no tractable alternative to unbounded iteration?
- Why do small {or unintended} changes in syntax produce wild changes in semantics instead of small ones {or syntax errors}?
- Why is program proof not mechanical?

No Science $\Rightarrow$ Better Mathematics

Programming language issues:

- Why does every language allow buffer overflow?
- Why is there no tractable alternative to unbounded iteration?
- Why do small {or unintended} changes in syntax produce wild changes in semantics instead of small ones {or syntax errors}?
- Why is program proof not mechanical?

We defined languages and programs; why don't we fix them?

Without Science Programmers are Cast Adrift

- Laws of the universe, according to Einstein, "...subtle, but not damn mean", are replaced by demands of a capricious client
- A structural engineer who says, "That's impossible" shows knowledge of reality; a software engineer who says the same confesses incompetence

"I will support those who say they can do it, not those who say they can't."

- Attributed to a star-wars project general

Do Labels Matter?

...that which we call a rose
By any other name would smell as sweet
...

Juliet didn't want Romeo's name to matter, but in the end their surnames killed them

- In a science the research methods and evaluation criteria are entirely different from those in mathematics
- Mathematics is more subjective
  But if elegance, simplicity, and generality are what's needed, computer science better start looking for them in its mathematics